

#3

1/10

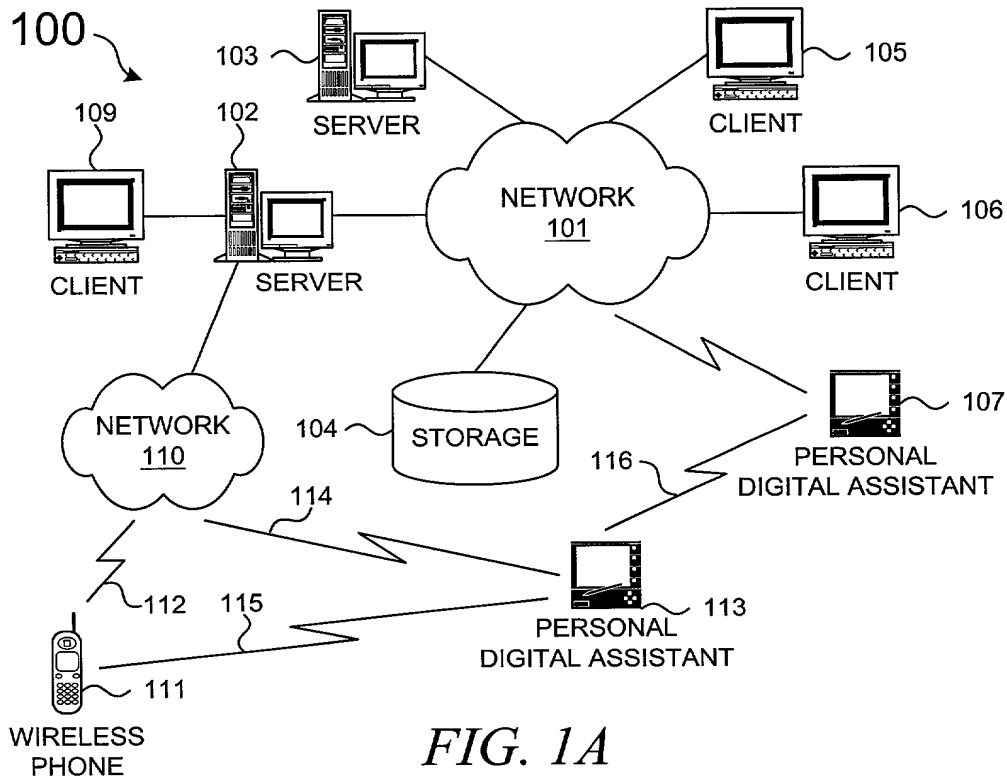


FIG. 1A
(PRIOR ART)

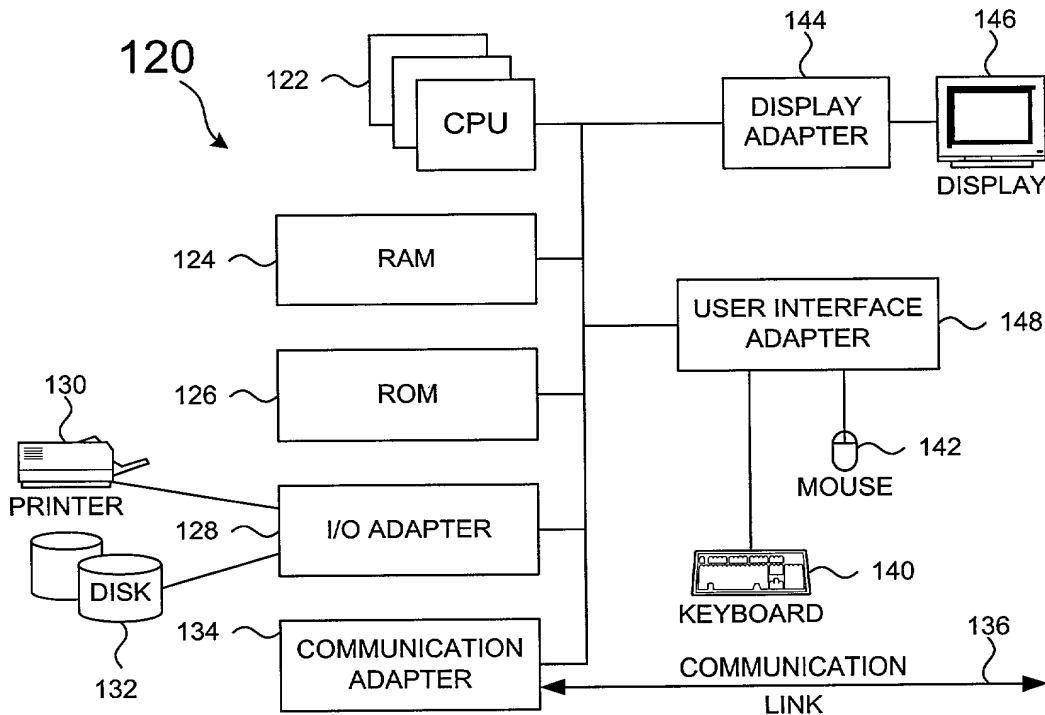


FIG. 1B
(PRIOR ART)

2/10

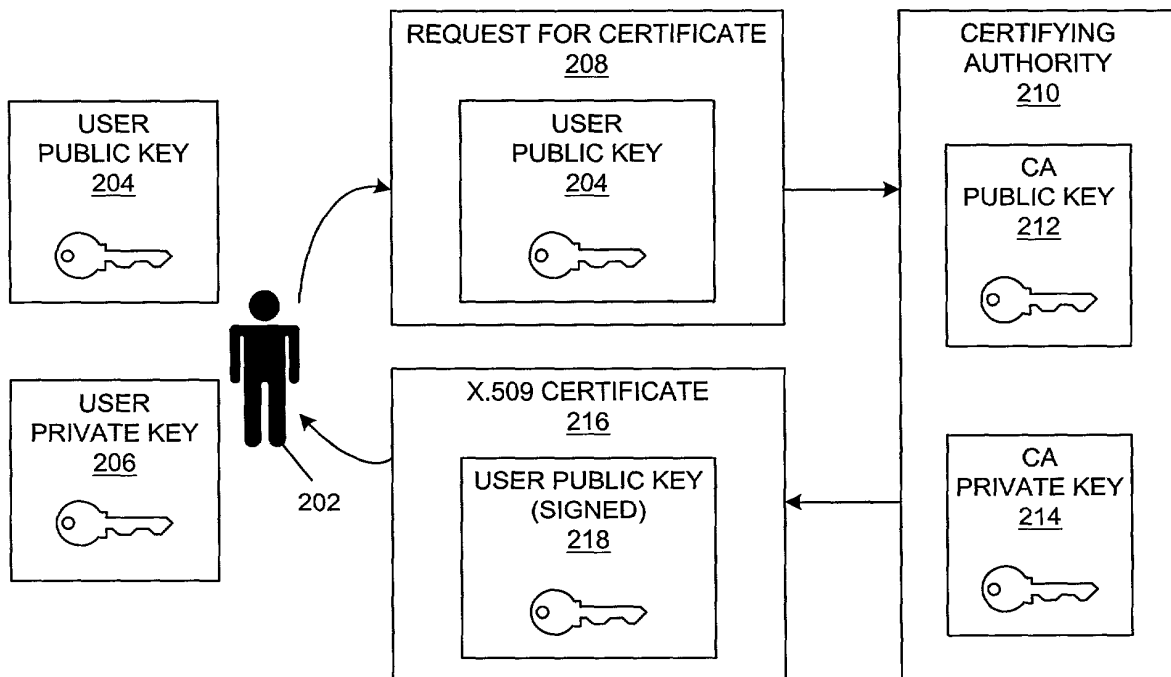


FIG. 2
(PRIOR ART)

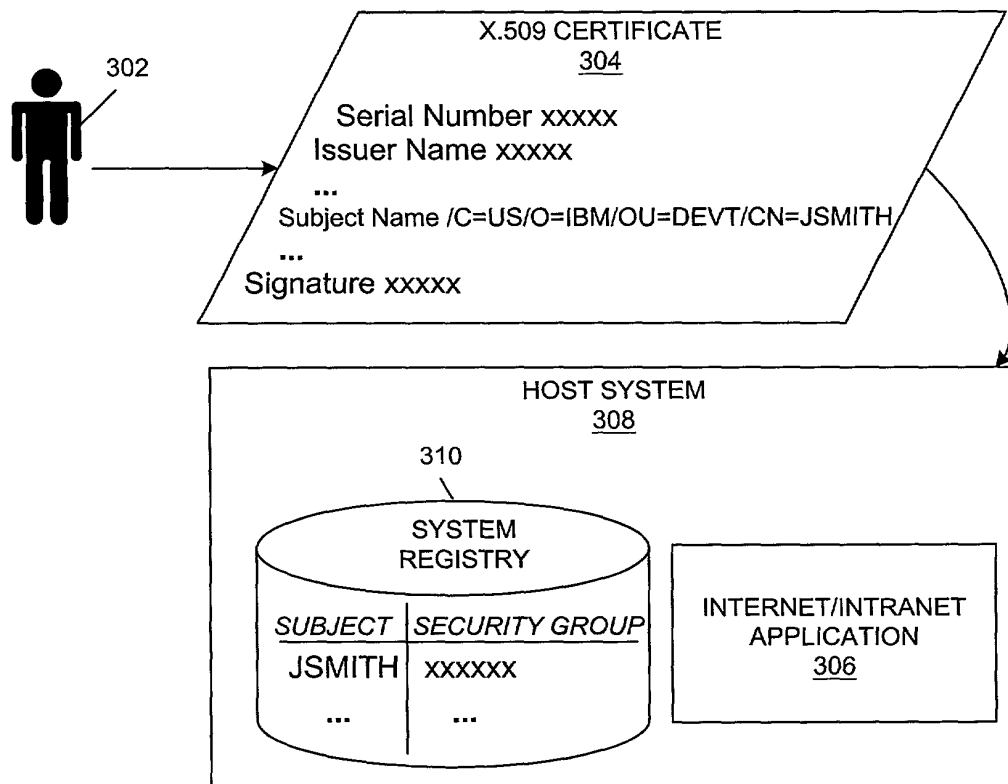


FIG. 3A
(PRIOR ART)

3/10

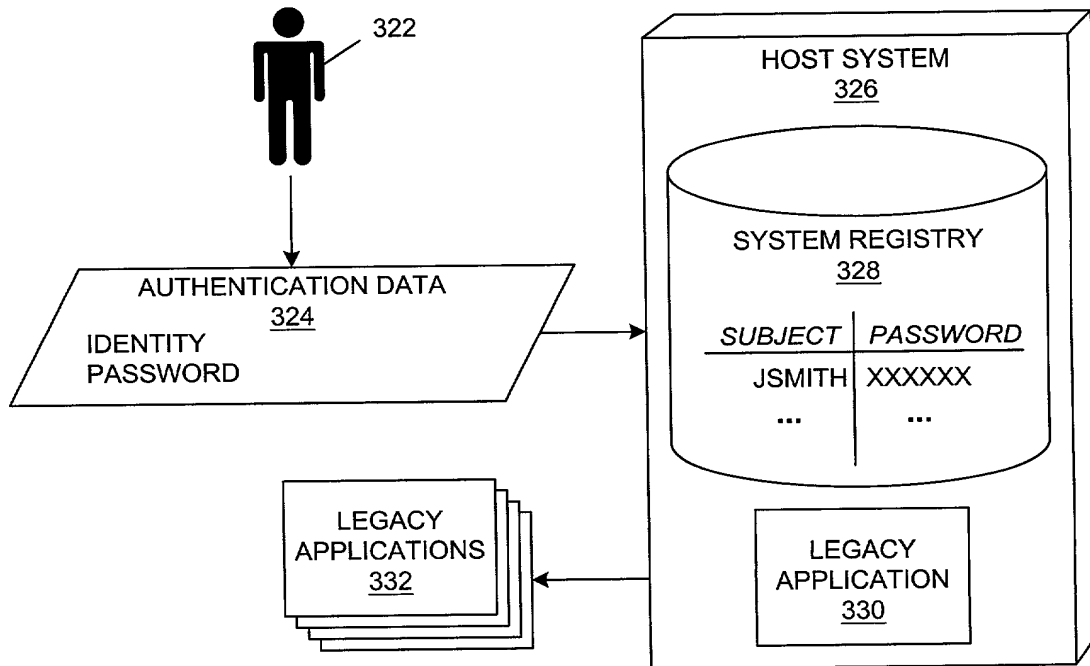


FIG. 3B
(PRIOR ART)

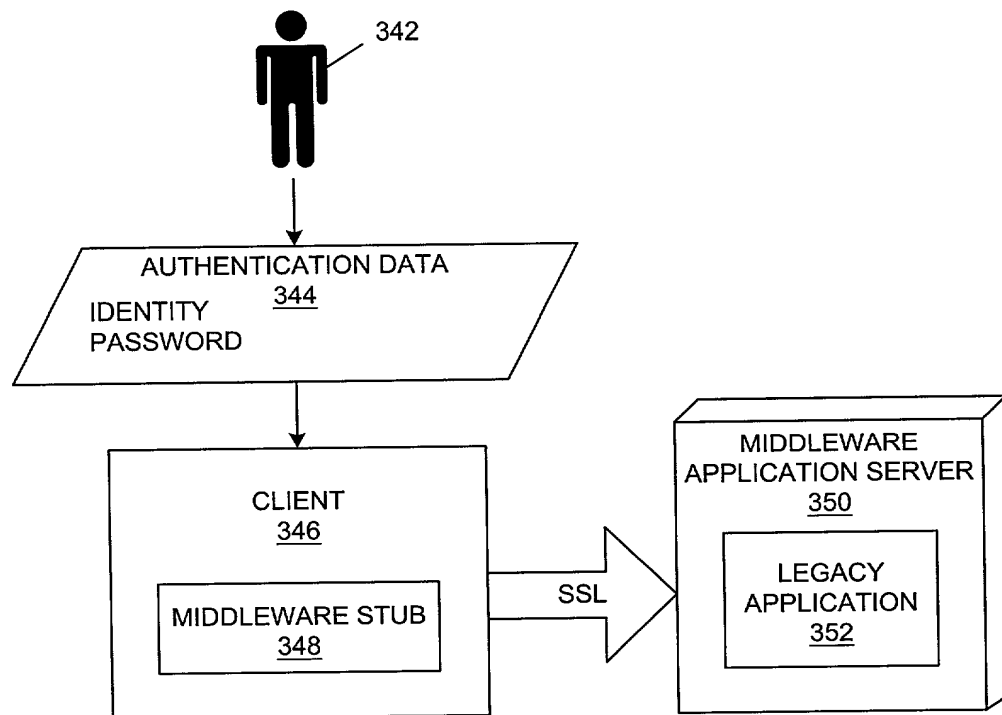


FIG. 3C
(PRIOR ART)

4/10

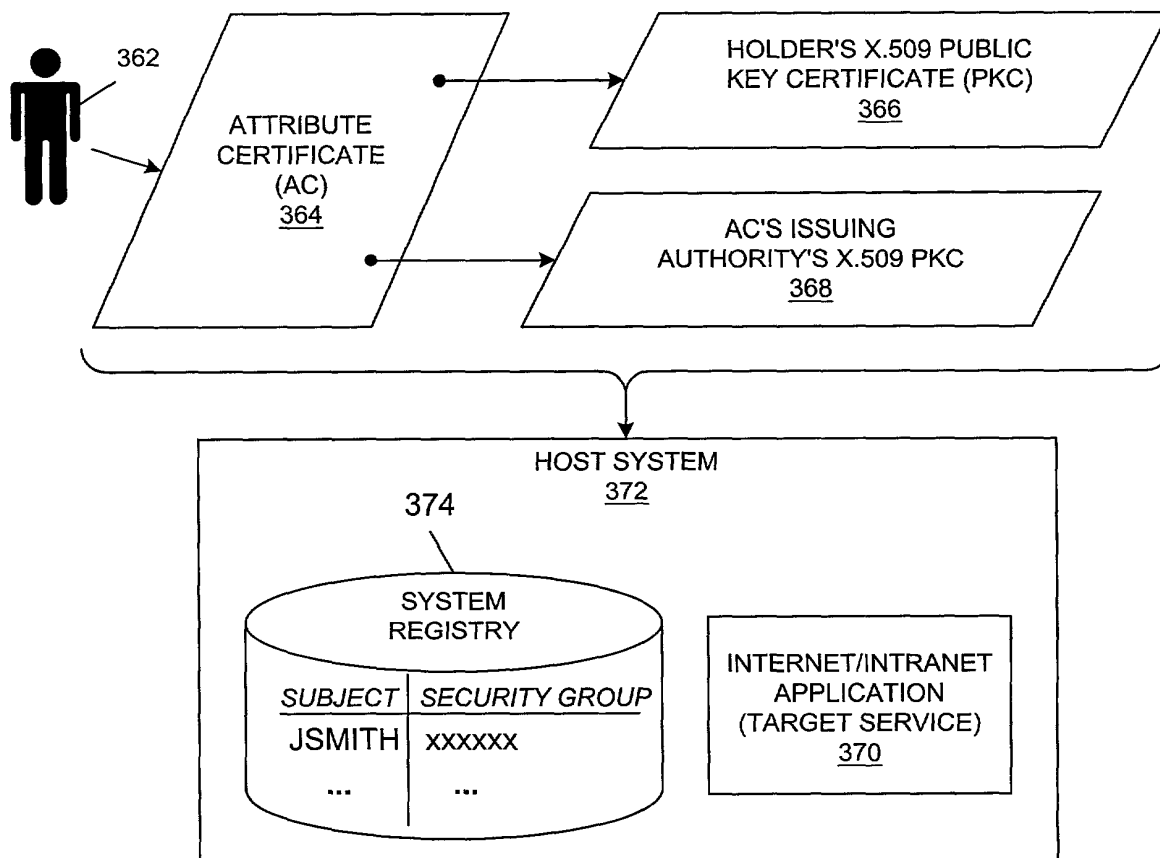


FIG. 3D
(PRIOR ART)

Method and system for public-key-based secure authentication to
distributed legacy applications

5/10

```

Certificate ::= SEQUENCE {
    tbsCertificate      TBSCertificate,
    signatureAlgorithm  AlgorithmIdentifier,
    signature           BIT STRING }

TBSCertificate ::= SEQUENCE {
    version             [0] Version DEFAULT v1,
    serialNumber        CertificateSerialNumber,
    signature           AlgorithmIdentifier,
    issuer              Name,
    validity            Validity,
    subject             Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    issuerUniqueID      [1] IMPLICIT UniqueIdentifier OPTIONAL,
    subjectUniqueID     [2] IMPLICIT UniqueIdentifier OPTIONAL,
    extensions          [3] Extensions OPTIONAL }

Version ::= INTEGER { v1(0), v2(1), v3(2) }

CertificateSerialNumber ::= INTEGER

Validity ::= SEQUENCE {
    notBefore          Time,
    notAfter           Time }

Time ::= CHOICE {
    utcTime            UTCTime,
    generalTime        GeneralizedTime }

UniqueIdentifier ::= BIT STRING

SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm           AlgorithmIdentifier,
    subjectPublicKey    BIT STRING }

Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension

Extension ::= SEQUENCE {
    extnID              OBJECT IDENTIFIER,
    critical            BOOLEAN DEFAULT FALSE,
    extnValue           OCTET STRING }

```

FIG. 4A

(PRIOR ART)

**Method and system for public-key-based secure authentication to
distributed legacy applications**

6/10

```

AttributeCertificate ::= SEQUENCE {
    acinfo          AttributeCertificateInfo,
    signatureAlgorithm AlgorithmIdentifier,
    signatureValue   BIT STRING
}

AttributeCertificateInfo ::= SEQUENCE {
    version          AttCertVersion DEFAULT v1,
    holder           Holder,
    issuer           AttCertIssuer,
    signature        AlgorithmIdentifier,
    serialNumber     CertificateSerialNumber,
    attrCertValidityPeriod AttCertValidityPeriod,
    attributes       SEQUENCE OF Attribute,
    issuerUniqueID   UniqueIdentifier OPTIONAL,
    extensions       Extensions OPTIONAL
}

AttCertVersion ::= INTEGER { v1(0), v2(1) }

Holder ::= SEQUENCE {
    baseCertificateID [0] IssuerSerial OPTIONAL,
    -- the issuer and serial number of
    -- the holder's Public Key Certificate
    entityName        [1] GeneralNames OPTIONAL,
    -- the name of the claimant or role
    objectDigestInfo   [2] ObjectDigestInfo OPTIONAL
    -- if present, version must be v2
}

ObjectDigestInfo ::= SEQUENCE {
    digestedObjectType ENUMERATED {
        publicKey          (0),
        publicKeyCert       (1),
        otherObjectTypes    (2) },
    -- otherObjectTypes MUST NOT
    -- be used in this profile
    otherObjectTypeID     OBJECT IDENTIFIER OPTIONAL,
    digestAlgorithm        AlgorithmIdentifier,
    objectDigest           BIT STRING
}

```

FIG. 4B
(PRIOR ART)

Method and system for public-key-based secure authentication to
distributed legacy applications

7/10

```

AttCertIssuer ::= CHOICE {
    v1Form          GeneralNames, -- v1 or v2
    v2Form          [0] V2Form    -- v2 only
}

V2Form ::= SEQUENCE {
    issuerName      GeneralNames OPTIONAL,
    baseCertificateID [0] IssuerSerial OPTIONAL,
    objectDigestInfo [1] ObjectDigestInfo OPTIONAL
    -- at least one of issuerName, baseCertificateID
    -- or objectDigestInfo MUST be present}

IssuerSerial ::= SEQUENCE {
    issuer      GeneralNames,
    serial      CertificateSerialNumber,
    issuerUID   UniqueIdentifier OPTIONAL
}

AttCertValidityPeriod ::= SEQUENCE {
    notBeforeTime GeneralizedTime,
    notAfterTime  GeneralizedTime
}

Attribute ::= SEQUENCE {
    type      AttributeType,
    values    SET OF AttributeValue
    -- at least one value is required
}

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY DEFINED BY AttributeType

```

FIG. 4C
(PRIOR ART)

```

name      id-aca-authenticationInfo
OID       { id-aca 1 }
Syntax    SvceAuthInfo
values:    Multiple allowed

```

```

SvceAuthInfo ::= SEQUENCE {
    service      GeneralName,
    ident        GeneralName,
    authInfo     OCTET STRING OPTIONAL
}

```

FIG. 4D
(PRIOR ART)

8/10

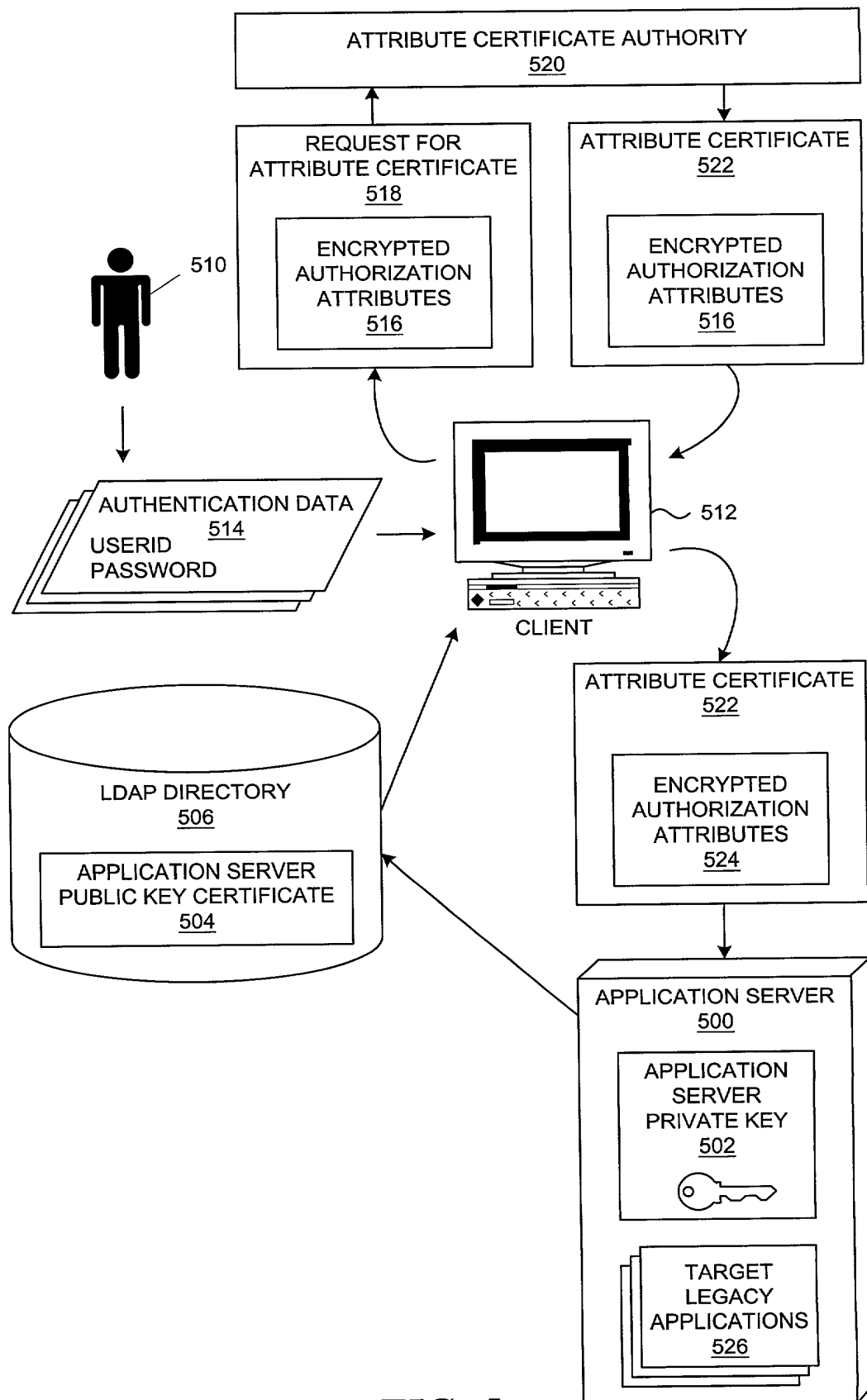


FIG. 5

9/10

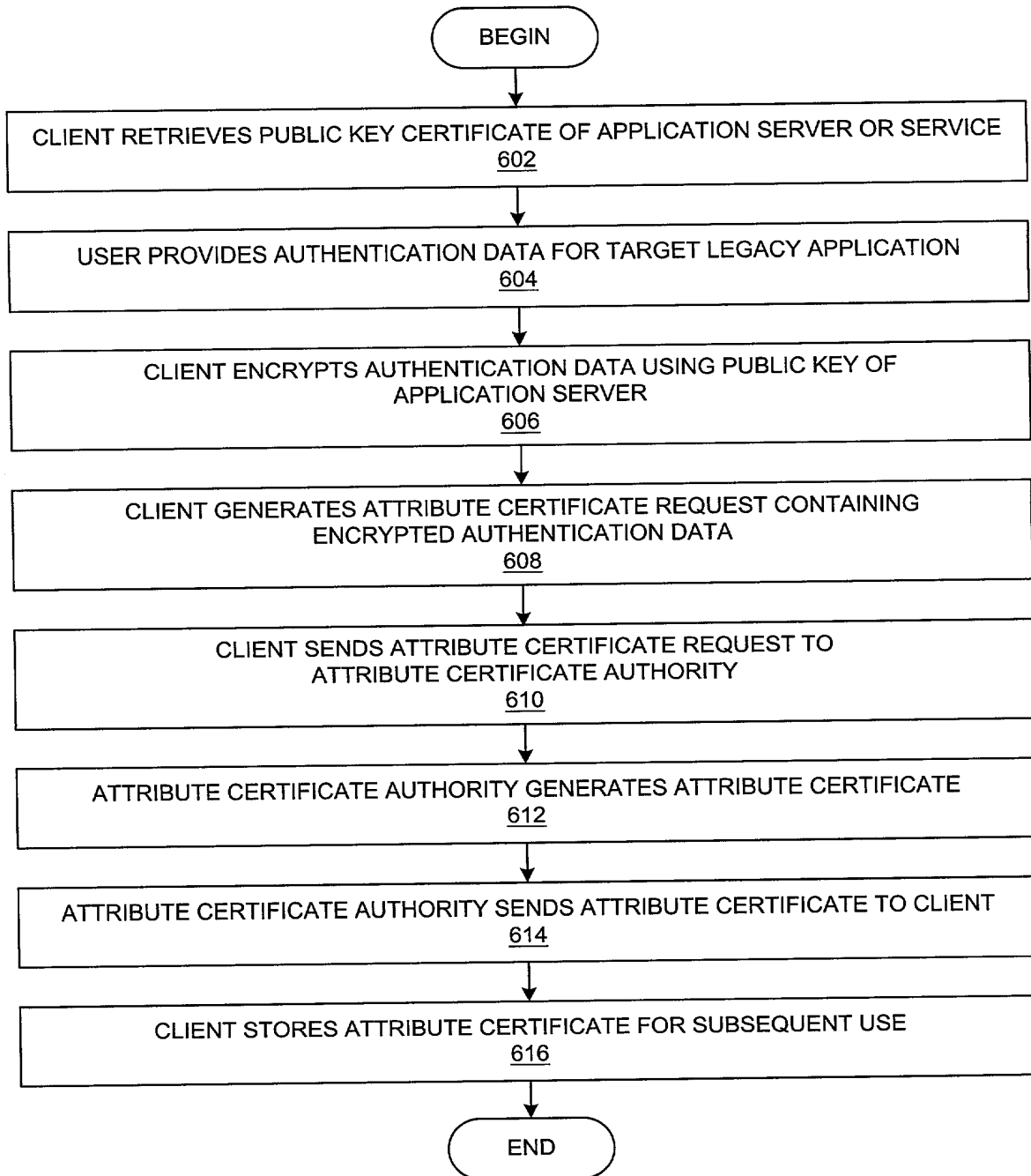


FIG. 6

10/10

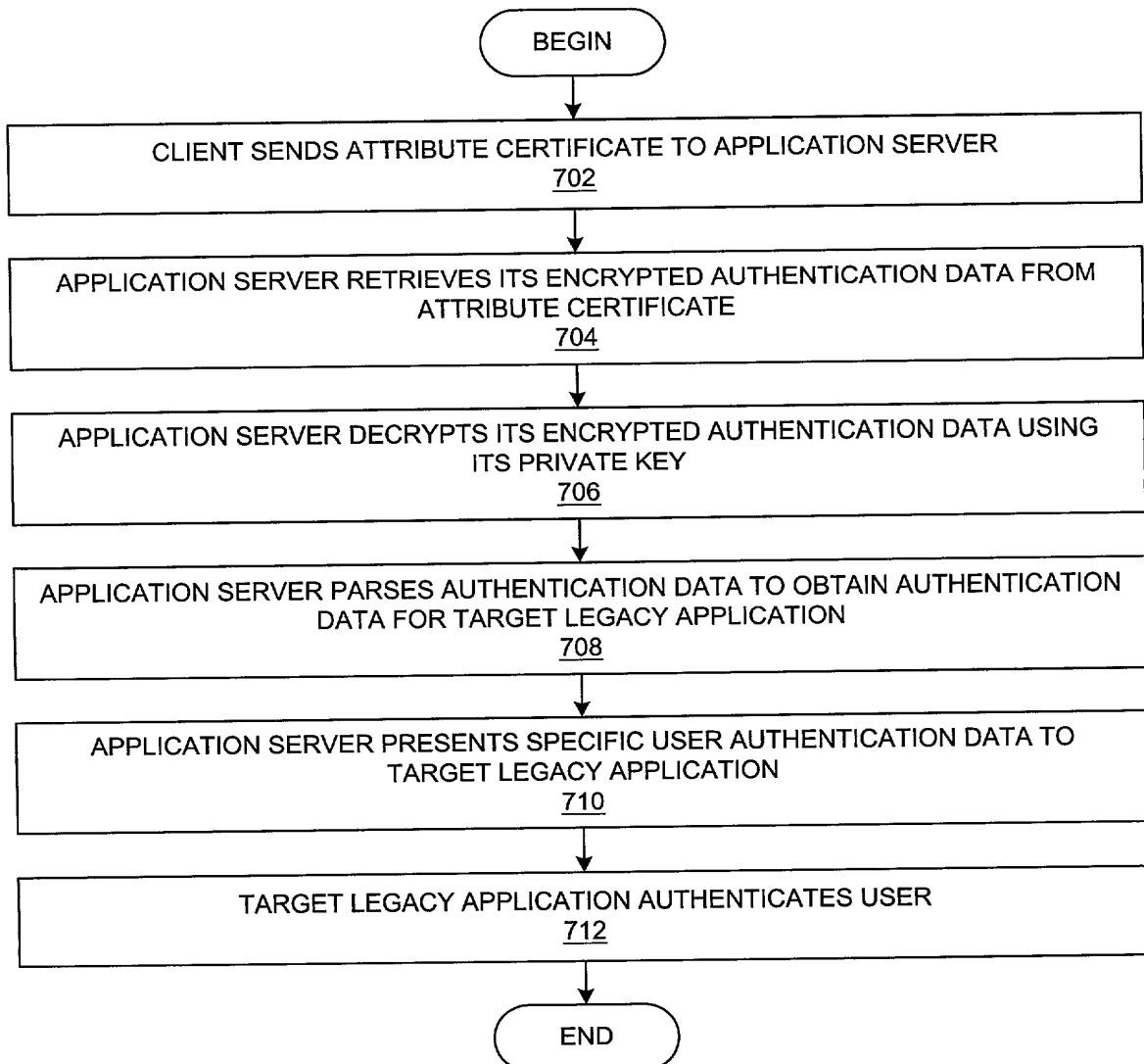


FIG. 7